

Important Questions About the "O" Words

by Anne Block, Technical Writer/Editor, Flashline

The "O" words — Open Source and Outsourcing — were a big deal in 2003. While neither is without controversy, each has seen increased interest — and [real world implementation](#). Both "O" words can save time and money in software development, but the success of your efforts depends on asking the right questions.

The Open Source Thing

Free, field-proven software assets? Sounds like a great deal. But you need to consider several things before delving into the Open Source world:

- How will you control which open source products are allowed and supported?
- How will you ensure cost-effective delivery of quality support?
- How will you track which projects use open source?
- How will you track which open source products are being used?
- How will you quantify savings from open source?
- How will you protect intellectual property ownership?

If you're doing the right thing in regard to Open Source assets, you'll always know What, Who, When, and Where. What asset was used? Who used it? When was it used? Where was it used (which project)? When you're collecting the information you need to answer those four questions, you're in a better position to answer the really big question: How do Open Source assets affect the bottom line?

Outsourcing: The Other "O"

At the same time that IT budgets shrank during the recent economic downturn, the demand for services continued apace. So what happens when your organization is at an all-time lean, mean strength and you still have projects that need to be completed, but you don't have the personnel in-house to make it happen?

Outsourcing is certainly a viable solution for resource gaps, but you need to understand that by using an outsourced resource, you shift a project's emphasis from development to contractor management. Some of the challenges that must be overcome with outsourcing include:

- Possible geographic dispersion
- Incompatible tools and techniques
- Cultural diversity
- Security classification differences
- Communication, teamwork, and coordination breakdowns

So the questions you must ask are:

- Are requirements and specifications clearly defined and readily available?
- Are you effectively communicating and enforcing architecture and design standards and goals?
- What are you doing to facilitate effective collaboration across locations and time zones?
- Have you taken steps to maximize (or *require*) the use of existing software assets?
 - Are corporate software assets secure?
 - What measures are in place to protect intellectual property?

As practical measures to ensure success with an outsourced project, you can add to your existing process with an [asset-centric](#) view of all outsourced development projects and the assets used and produced in the course of the projects. In addition, dynamic notification, feedback, and discussion forums will help to bridge gaps between teams. The increased visibility of your dynamic asset inventory enables internal and external teams to easily work together. Tool-independent technology spans development environments so that you can discover similar products being developed in different projects and collaborate between them. Each of these items is eminently possible within an enterprise.

Open Source software and outsourcing are tools, not ends in themselves. Each offers the potential to have a significant positive impact on the bottom line — but only if you ask the right questions. We've shown you some of the considerations. Now it's up to you to conquer the challenges of the "O" words and make them work for you in an agile, practical development environment.

